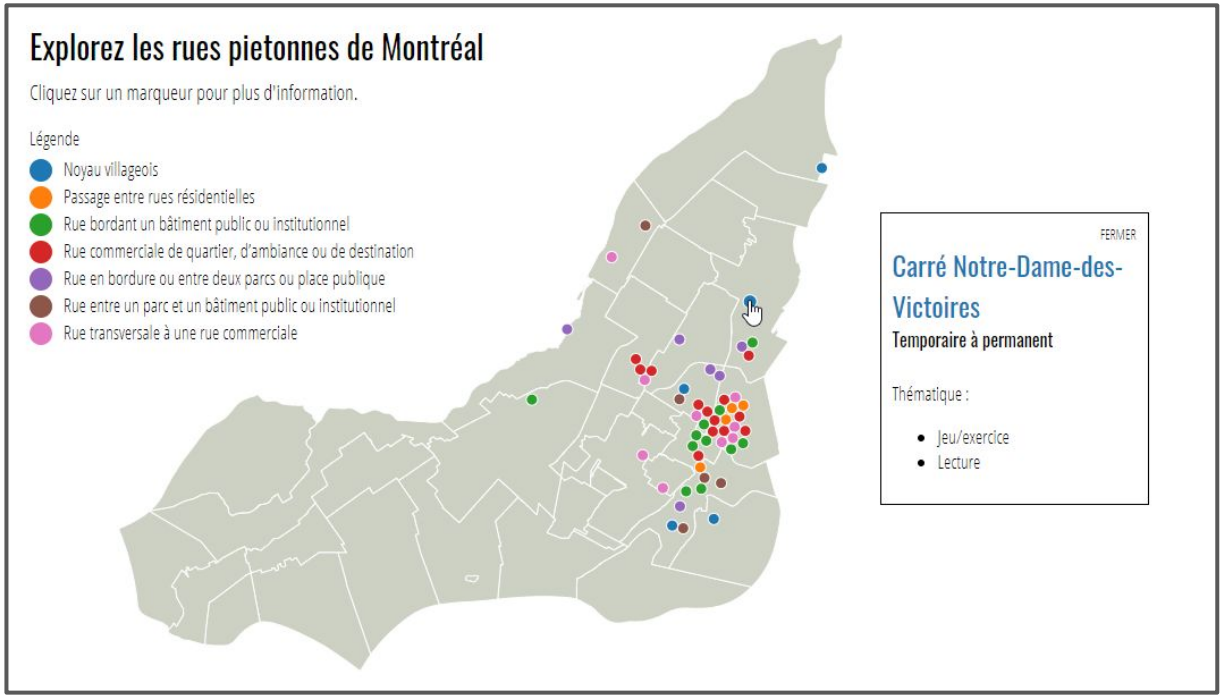
Guide TP5

INF8808 | Summer 2022

Version JavaScript

Objective

- The objective of this TP is to create an interactive scatter map using open data in JSON and GeoJSON formats.



Data

Pedestrian streets of Montreal

- The data represent the pedestrian streets of Montreal, as well as the geographic limits of each borough of Montreal
- `src/assets/data/`
- The pedestrian streets dataset contains :
  - **MODE_IMPLANTATION**
  - **NOM_PROJET**
  - **OBJECTIF_THEMATIQUE**
  - **TYPE_SITE_INTERVENTION**
- The geographic dataset contains:
  - **NOM**
  - **CODEID**

Data preprocessing

Purpose: *Reorganize some parts of the data so that they can be more easily used with D3.js*

In `preprocessing.js` :

1. ***convertCoordinates***

2. ***simplifyDisplayTitles***

```
{
    "type":"FeatureCollection",
    "crs":{...},
    "features":[
        {
            "type":"Feature",
            "properties":{

"ID_PROJET":"RP0001",

"TYPE_AXE":"Avenue", ...
            },
            "geometry":{...},
            "x":523.9884058026364,
            "y":287.12224173489085
        }, ...
    ]
```

Scatter map

We must draw Montreal and its neighborhoods as well as the markers

In `viz.js` :

1. ***colorDomain***
2. ***mapBackground*** and ***showMapLabel***
3. ***mapMarkers***

Scatter map

Remarks

- When a neighborhood is hovered over by the mouse, its name appears in the center of its polygon
  - `path.centroid`
- The colors of the dots depend on the type of site they represent
- Circles appear with a white outline and increase in size slightly when hovered
- The circles should automatically move so that there are no overlaps between them
  - `helper.js`

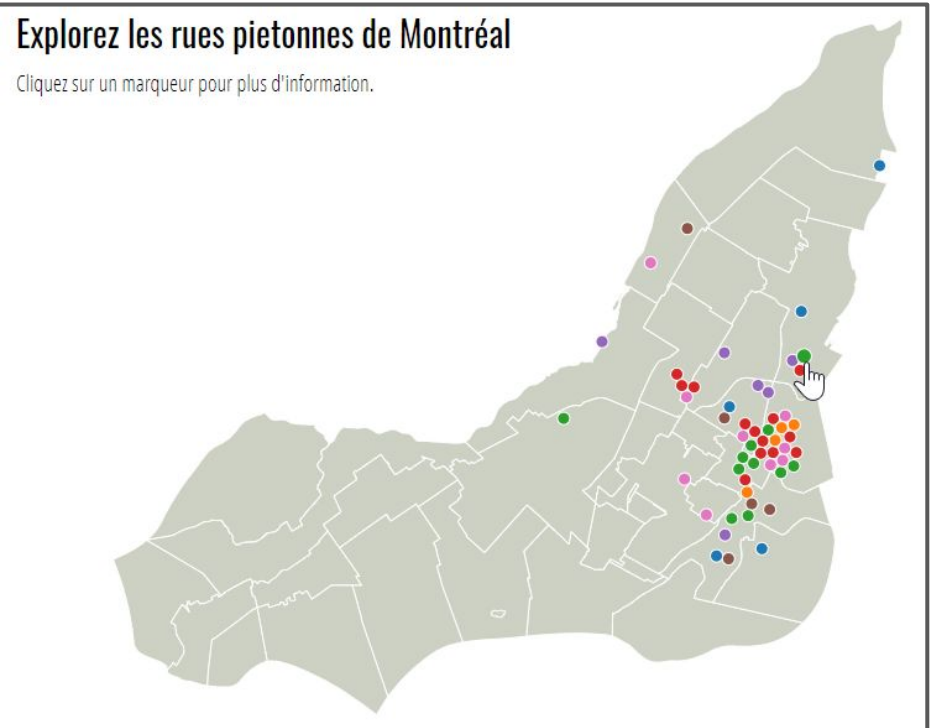
```
/**
 * Initializes the simulation used to place the circles
 *
 * @param {object} data The data to be displayed
 * @returns {*} The generated simulation
 */
export function getSimulation (data) {
  return d3.forceSimulation(data.features)
    .alphaDecay(0)
    .velocityDecay(0.75)
    .force('collision',
      d3.forceCollide(5)
        .strength(1)
    )
}

/**
 * Update the (x, y) position of the circles'
 * centers on each tick of the simulation.
 *
 * @param {*} simulation The simulation used to position the cirles.
 */
export function simulate (simulation) {
  simulation.on('tick', () => {
    d3.selectAll('.marker')
      .attr('cx', (d) => d.x)
      .attr('cy', (d) => d.y)
  })
}

/**
```

helper.js: contains functions that move circles to avoid collisions

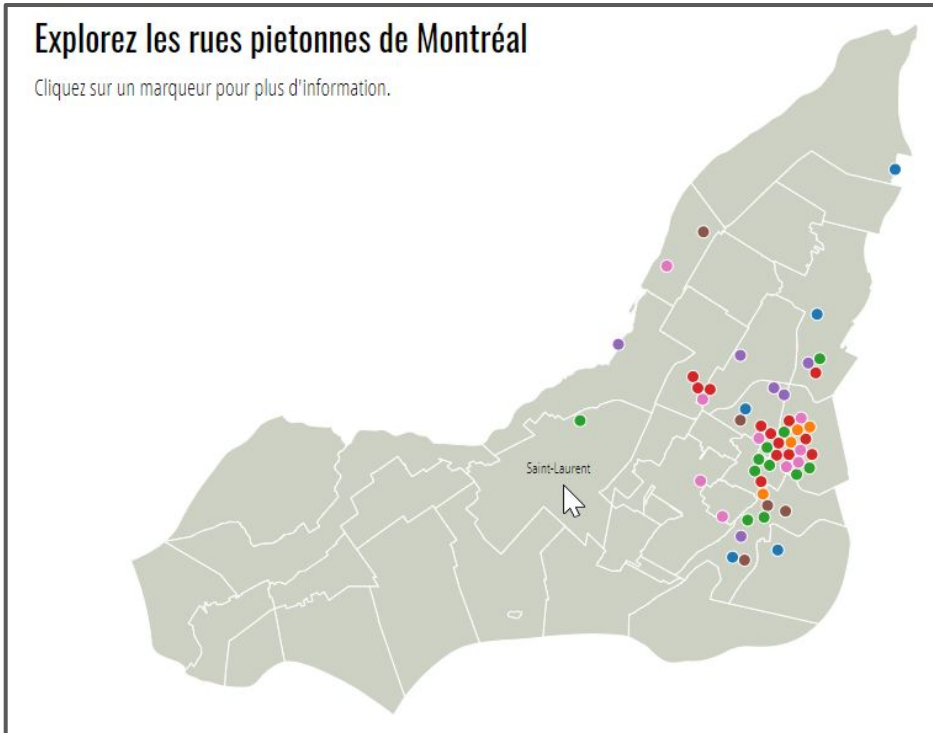
Scatter map

Initial result

- **Each circle gets bigger on hover:**



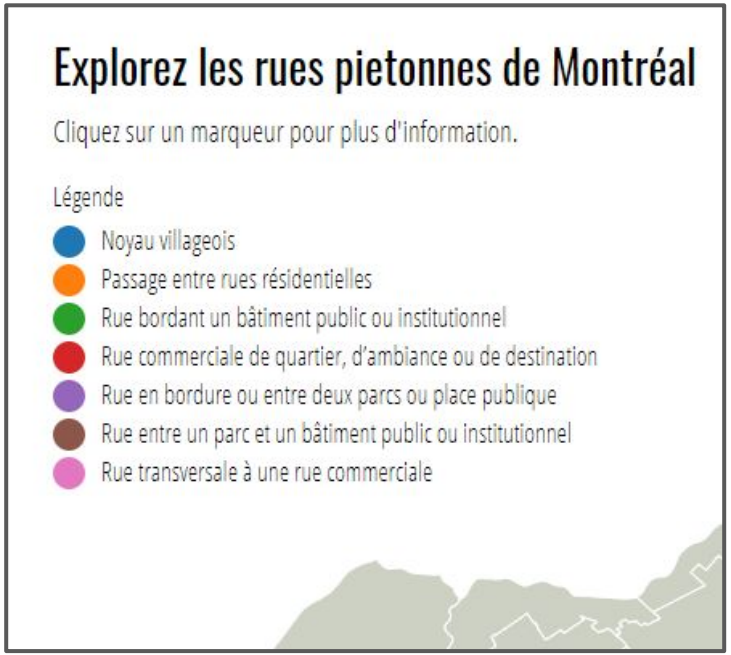
- **Neighborhood name appears in its center :**



Legend

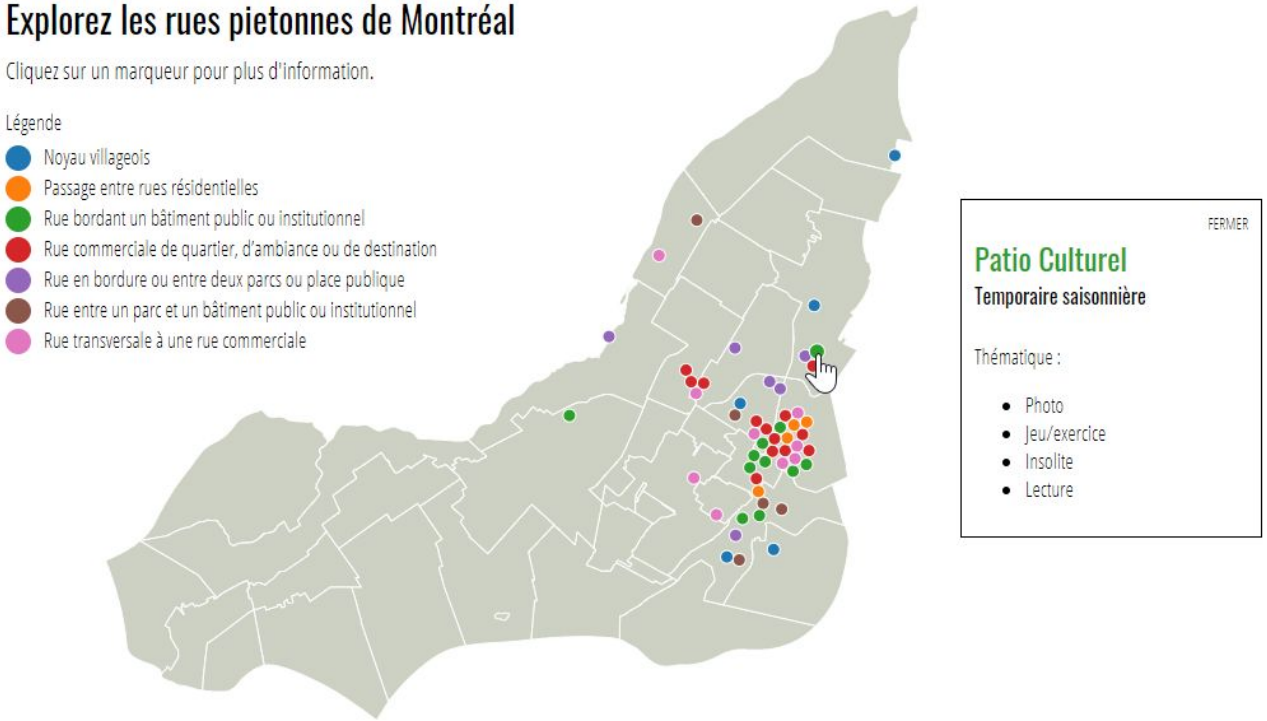
- `legend.js`
- Like in TP4, we use the library ‘`d3-svg-legend`’

Result :



Information panel

Illustrated



Information panel

- A panel giving information on a given point should be displayed on the right when clicked
- `hover.js` :

    1. ***setTitle***
    2. ***setMode***
    3. ***setTheme***

Submission date: June 12th 11:59PM